

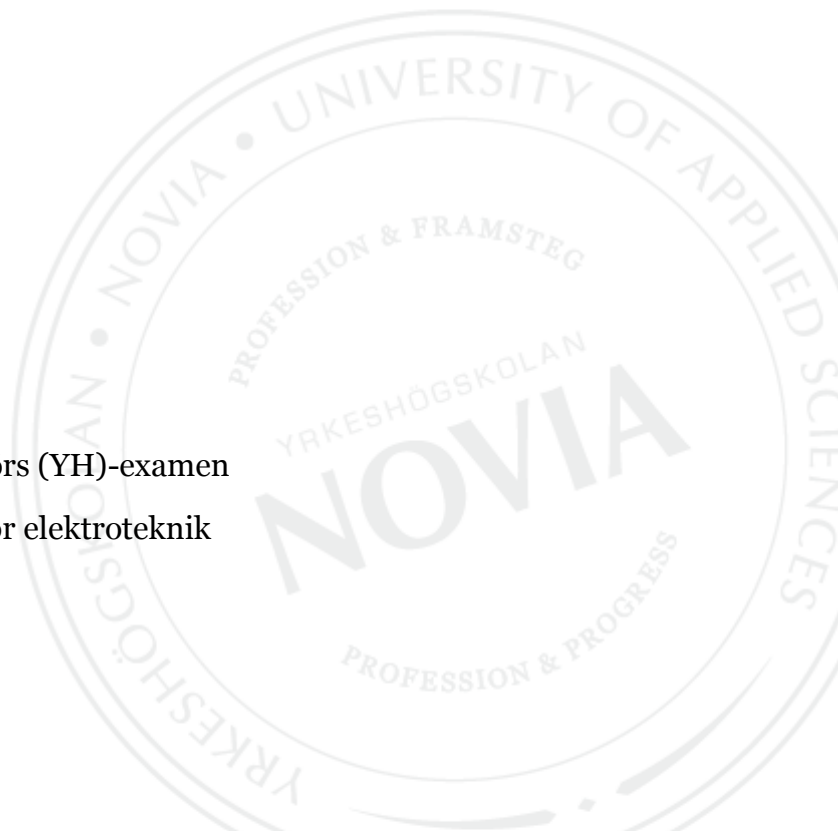
Design av belastningstestapplikation för MIRKA® DEROS 5650CV slipmaskin

Jakob Staffans

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för elektroteknik

Vasa 2015



EXAMENSARBETE

Författare: Jakob Staffans

Utbildningsprogram och ort: Elektroteknik, Vasa

Inriktningsalternativ: Automationsteknik

Handledare: Roger Mäntylä

Titel: *Design av belastningstestapplikation för MIRKA® DEROS 5650CV slipmaskin*

Datum 6.2.2015	Sidantal 34	Bilagor 1
----------------	-------------	-----------

Abstrakt

Detta examensarbete gjordes åt Power Tools avdelningen på KWH Mirka Ab. Målet var att skapa en belastningstestapplikation för elslipmaskiner för att enkelt och definitivt kontrollera maskinernas arbetstemperaturer. Applikationen är tänkt att användas i Mirkas after-sales enheter runtom i världen och måste därför vara användarvänlig.

Programmeringen gjordes med Visual Basic .NET. Därefter utvärderades testprocessens tillförlitlighet med ANOVA Gage R&R. Testprocessen bedömdes som tillförlitlig och användes därmed för att bestämma sina egna temperaturgränsvärden genom testkörningar av slipmaskiner.

Språk: svenska	Nyckelord: Mirka, Visual Basic .NET, ANOVA Gage R&R
----------------	---

OPINNÄYTETYÖ

Tekijä: Jakob Staffans

Koulutusohjelma ja paikkakunta: Sähkötekniikka, Vaasa

Suuntautumisvaihtoehto: Automaatiotekniikka

Ohjaajat: Roger Mäntylä

Nimike: *MIRKA® DEROS 5650CV -hiomakoneen kuormitustestin ohjelmointi*

Päivämäärä 6.2.2015 Sivumäärä 34 Liitteet 1

Tiivistelmä

Tämä opintotyö tehtiin KWH Mirka Oy:n Power Tools -osastolle. Pää tavoitteena oli luoda testiohjelma, joka määrittelee sähköhiomakoneiden kuormitusta ja helposti ja tehokkaasti tarkistaa koneiden työlämpötiloja. Ohjelma on tarkoitettu käytettäväksi Mirkan after-sales-yksiköissä ympäri maailmaa ja siksi sen pitää olla käyttäjäystävällinen.

Ohjelmointi tehtiin Visual Basic .NET -kielellä. Sen jälkeen arvioitettiin testitapahtumasarjan luotettavuus ANOVA Gage R&R -välineellä. Testiprosessi arvioitiin luotettavana, minkä takia sitä käytettiin määriteltäessä testiprosessin omat lämpötilaraja-arvot hiomakoneiden koeajojen avulla.

Kieli: ruotsi

Avainsanat: Mirka, Visual Basic .NET, ANOVA Gage R&R

BACHELOR'S THESIS

Author: Jakob Staffans

Degree Programme: Electrical Engineering, Vasa

Specialization: Automation

Supervisors: Roger Mäntylä

Title: *Designing of load test application for MIRKA® DEROS 5650CV sander*

Date 6.2.2015 Number of pages 34 Appendices 1

Summary

This Bachelor's thesis was done for the Power Tools department at KWH Mirka Ab. The goal was to create a load test application for their electric sanders in order to monitor the working temperatures in a simple and definitive way. This application is meant to be used at Mirka's After-Sales units around the world and therefore it has to be user-friendly.

The programming was done with Visual Basic .NET. Afterwards the reliability of the testing process was evaluated using ANOVA Gage R&R. The testing process was considered reliable and was hence utilized to determine its own temperature limits by test runs of electric sanders.

Language: Swedish Key words: Mirka, Visual Basic .NET, ANOVA Gage R&R

Innehållsförteckning

1	Inledning.....	1
1.1	Syfte och mål.....	1
2	KWH Group Ltd.....	2
2.1	KWH Mirka Ab.....	2
2.2	KWH Logistics Ab.....	2
2.3	KWH Invest Ab.....	2
3	Problemformulering	3
4	Teoretisk utgångspunkt.....	4
4.1	MIRKA® DEROS 5650CV	4
4.2	Bromsfläkt	5
4.3	Tidigare Mirka-applikation	5
4.3.1	Textloggningsprogram.....	6
4.4	Minitab	6
4.5	ANOVA Gage R&R.....	7
5	Visual Basic .NET	8
5.1	Objektorienterad programmering	8
5.2	Klasser	9
5.3	Timer	9
6	Metoder och utförande.....	10
6.1	Påbörjande	10
6.2	Funktionsprincip.....	10
6.3	Problem med Bluetooth-kommunikation	12
6.4	Datatolkning och testtid.....	13
6.5	Felhantering, kompatibilitet och underhåll	15
7	Användbarhet och användarvänlighet.....	16
7.1	Definitioner	16
7.2	Klarhet och spårbarhet.....	17
7.3	Konsistens	18
8	ANOVA Gage R&R testresultat	20
8.1	Resultat	20
9	Gränsvärden	23
10	Resultat	26
11	Diskussion	27

Figurförteckning

Figur 1. MIRKA® DEROS 5650CV	4
Figur 2. Ramverket för testapplikationen	5
Figur 3. Exempel på flaggor	6
Figur 4. Exempel på en klass	9
Figur 5. Blockschema för kommandoloopen	11
Figur 6. Load Test, belastningstestapplikationen	11
Figur 7. Flödesschema för testprocessen	12
Figur 8. Exempel på en timer	13
Figur 9. En bristfällig tidig version av det grafiska gränssnittet	17
Figur 10. Den slutliga versionen av det grafiska gränssnittet.....	19
Figur 11. En tabell av kretskortens temperaturskillnader i ANOVA-testet	20
Figur 12. En tabell av temperaturerna hos motorerna i ANOVA-testet	21
Figur 13. Analys av kretskortstemperaturerna med Minitab.....	21
Figur 14. Analys av motortemperaturerna med Minitab	22
Figur 15. Histogram på temperaturen på 20 oinkörda slipmaskiner	23
Figur 16. Ekvation för att bestämma standardavvikelsen	24
Figur 17. Ekvation för att bestämma gränsvärden	24
Figur 18. Normalfördelning	25

1 Inledning

Detta examensarbete gjordes för avdelningen Power Tools på KWH Mirka Ab i Jeppo, Nykarleby. Power Tools planerar, designar och producerar handhållna slipmaskiner. Uppgiften var att programmera en testapplikation för utvärdering av temperaturändringar i slipmaskiner.

1.1 Syfte och mål

Syftet med att skapa detta program är att det ska kunna användas vid Mirkas after-sales service center för felaktiga slipmaskiner. Maskinen ska snabbt kunna köras genom testapplikationen, motta temperaturdata och analysera denna för att se om slipmaskinen överhettas och således är felaktig.

Målet var att programmera en fungerande första version av testapplikationen och utforma ett lämpligt grafiskt gränssnitt. Människor i andra länder ska lätt kunna använda programmet. Därför beaktades användarvänlighet i utformningen och en manual med användarinstruktioner skapades också. Efteråt användes ANOVA Gage R&R-metoden för att utvärdera om själva testprocessen är tillförlitlig. Till slut diskuteras resultaten och vidareutveckling föreslås.

2 KWH Group Ltd

Mirka är del av KWH Group, som består av KWH Mirka, KWH Logistics och KWH Invest. KWH Group var ursprungligen ett timmerbolag som startades av Emil Höglund och Edwin Wiik 28:e augusti 1929. Företaget Wiik & Höglund kom att arbeta med timmer- och massavedsexport, plastproduktion, slipmaterial med mera. År 1937 startade Karl Johan Stuns och Emil Höglund aktiebolaget Keppo för att idka minkfarmning. Med inkomsterna därifrån köpte Höglund majoriteten av aktierna i slipmaterials företaget Mirka. Med den affären föddes KWH Group, i vilken Mirka kom att växa till en av grundpelarna. Bolaget gav dotterbolagen mycket självständighet och blev således mer av en investerare. (KWH Group Ltd., 2009)

2.1 KWH Mirka Ab

Mirka är en tillverkare av högkvalitativa slipmaterial och slipmaskiner. Det är ett globalt växande företag med dotterbolag i Europa, Nord- och Sydamerika och Asien. Högkvarteret och produktionen ligger i Finland. I Jeppo finns Power Tools avdelningen, där handhållna elektriska slipmaskiner utvecklas. Där byggdes nyligen också ett teknologi center, i vilket Mirka tränar kunder och försäljare samt utvecklar produkter. (KWH Group Ltd., 2009)

2.2 KWH Logistics Ab

KWH Logistics består av två dotterföretag: KWH Freeze – kallförvaring och Backman-Trummer Group som har hand om hamnarbete och sjöfrakt, frakt på land och industriella tjänster. KWH Logistics Ab säljer kundspecifika logistiklösningar. (KWH Group Ltd., 2009)

2.3 KWH Invest Ab

KWH Invest Ab består av Prevex och Uponor Infra Ltd. Prevex gör injektionsformade HVAC-produkter, alltså produkter för upphettning, ventilation och luftkonditionering samt andra kundspecifika nischprodukter. Prevex är marknadsledare i Skandinavien för uppsugningssystem för diskhoar. Uponor Infra utvecklar, producerar och säljer plaströr för industriella och kommunala tillämpningar samt tillämpningar i byggnadsbranschen. (KWH Group Ltd., 2009)

3 Problemformulering

För att lättare förstå vad som är fel i felaktiga slipmaskiner bestämdes det att ett belastningstestprogram skulle skapas. Det fanns redan en applikation som möjliggjorde kommunikation mellan en PC och en slipmaskin. I applikationen syntes temperatur och strömmar på alla faser i slipmaskinens synkronmotor, temperaturer på kretskortet och motorn, deltaström med mera.

Uppgiften var att med denna applikation som ramverk, skapa en ny applikation som skulle köra en slipmaskin ett visst tidsintervall, logga dess temperaturer vid olika ögonblick och sedan ge ett svar om slipmaskinen är felaktig eller om den klarar testet. Om maskinen hamnat på reparation är måhända någonting fel ändå, men med detta test kan överhettning uteslutas eller konstateras. Analysen av datainsamlingen var en del av problemet. Temperaturhöjningen var inte linjär så det behövde övervägas vilka effekter det har.

Den tidigare applikationen åskådliggjorde aktuella strömmar och temperaturer men hade dåliga möjligheter för synkroniserad loggning och körning av slipmaskiner. Utan det skulle det vara svårt att dra slutsatser om maskiner. I testapplikationen som skapades körs slipmaskinerna en bestämd tid via programmet, vilket skapar precision. Data loggas för analys och senare användning, samt åskådliggörs grafiskt. Till slut gjordes också en användarmanual för testapplikationen.

Efter att testapplikationen konstaterats fungera användes ANOVA Gage R&R-metoden för att utvärdera testprocessen. Detta gjordes för att få vetskap om huruvida testet är tillförlitligt, om det varierar mellan personer och om samma maskin ger närapå samma värde vid flera testkörningar. Efter processen konstaterats vara pålitlig bestämdes temperaturgränsvärden som slipmaskinerna bör klara av för att utesluta överhettning.

4 Teoretisk utgångspunkt

För att kunna skapa testapplikationen behövdes först Mirkas slipmaskin studeras. Slipmaskinen i fråga var MIRKA® DEROS 5650CV. För att simulera slipning användes en specialgjord bromsfläkt på slipmaskinen istället för den normala fästplattan för sandpapper. Därtill studerades en tidigare Mirka applikation noggrant för att få en uppfattning om hur Mirkas applikationer är uppbyggda. Slutligen utvärderades testprocessen med ANOVA Gage R&R, som är ett verktyg för mätsystemanalys.

4.1 MIRKA® DEROS 5650CV

MIRKA® DEROS 5650CV är en av Mirkas nyaste slipmaskiner. Det är en 1050 g lätt slipmaskin i plasthölje med en fem eller sex tums fästplatta för sandpapper och en 350 W borstlös synkronmotor med sex faser. Ovanpå slipmaskinen finns ett ställbart handtag som reglerar farten på motorn från fyra- till tiotusen varv per minut, dessutom har den en inbyggd elektronisk motorbroms. En utsugskanal finns inbyggd till vilken en dammsugare kopplas, vilket ger dammfri slipning. Den har en av-och-på knapp och två knappar för att öka respektive sänka farten.

Inuti höljet finns temperaturgivare för kretskortet och motorn, vilka kom att spela roll i detta slutarbete. Med hjälp av BlueTooth kan de kontaktas från datorn, läsa data och styra motorn. (KWH Mirka Ltd., 2013)



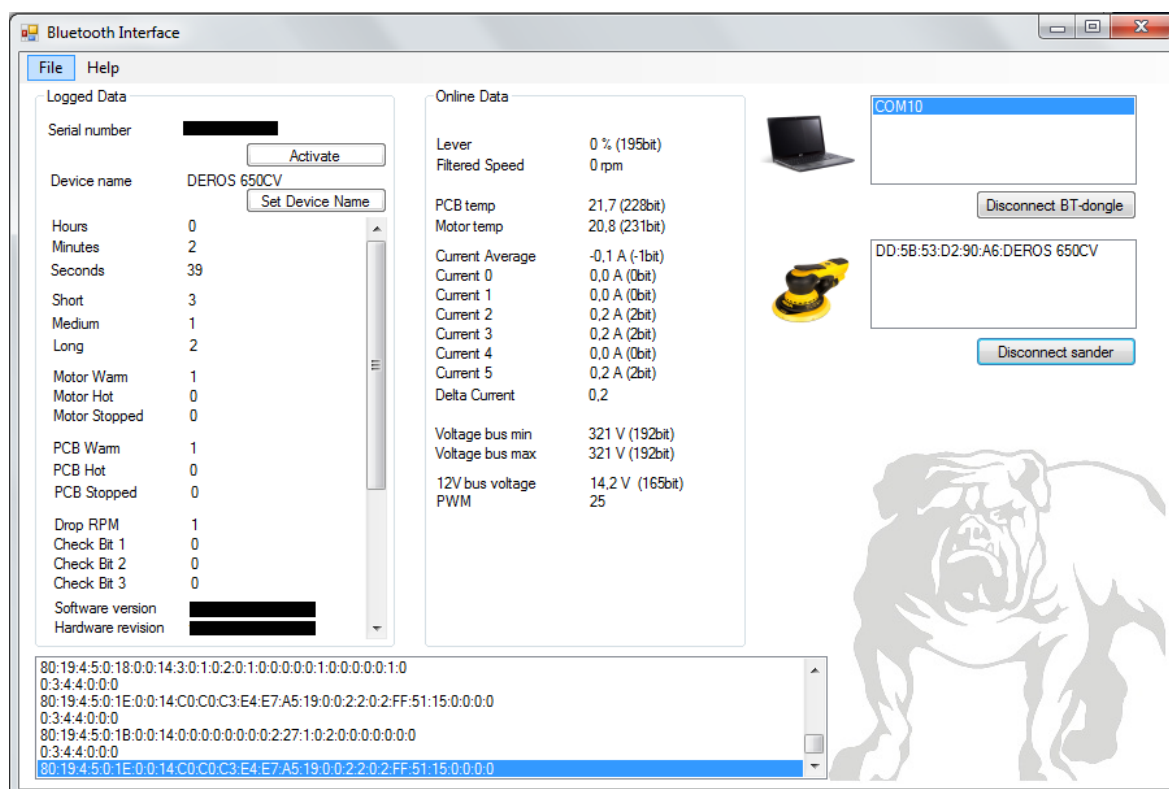
Figur 1. MIRKA® DEROS 5650CV som användes i examensarbetet.

4.2 Bromsfläkt

För att göra testet standardiserat och samtidigt underlätta arbetet för operatören har det tidigare designats en bromsfläkt som skulle simulera bromskraften på motorn som uppkommer när en yta slipas. Fästplattan för sandpapper byts helt enkelt ut mot en plastfläkt som är designad att med hjälp av luftflöde bromsa så mycket som möjligt. Märkbar skillnad märktes när maskinen kördes i tre minuter utan fläkt och nådde 44 °C, jämfört med när fläkten användes – cirka 100 °C. (Personlig kommunikation, Södergård, 6.6.2014)

4.3 Tidigare Mirka-applikation

Enheten Power Tools har en egenhändigt gjord omfattande applikation för diagnostisering av slipmaskiner. Till det hör också en egen drivrutin för BlueTooth-adaptern, som möjliggör anslutning mellan en dator och en slipmaskin.



Figur 2. Ramverket för testapplikationen. Mirkas diagnostikprogram för slipmaskiner.

I detta Mirkas diagnostikprogram visas data för slipmaskinen: temperatur på kretskortet och i motorn, strömmar på motorns alla sex faser, deltaström, hård- och mjukvaruversioner, serienummer, slipmaskinsmodell med mera.

Även ett par andra program fanns tillgängliga för att användas som mallar för kodningen av testapplikationen. Ett textloggningsprogram var till stor hjälp. Ur dessa kunde funktionsprinciper för program i allmänhet förstås men även kodsegment och funktioner. (Personlig kommunikation, Södergård, 6.6.2014)

4.3.1 Textloggningsprogram

Ett loggningsprogram fanns tillgängligt från förr med vilket det kunde loggas strömmar och temperaturer för senare användning i ett Text- eller Exceldokument. Upplägget för programmet var klart, men det behövde ställas in vilka värden som skulle fås ut, dessutom skulle det integreras i belastningsapplikationen med hjälp av flaggor och kommandon så att det skulle fungera i samverkan med belastningstestet. (Personlig kommunikation, Södergård, 10.7.2014)

```
If TxtLogBtn.Text = "Start logging" Then           // Flagga för start
    TxtLogBtn.Text = "Stop logging"               // Återställ flaggan
    LogTimer.Enabled = True                       // Startar loggningstimern, ändrar Enabled-attributet
    ...

Else If TxtLogBtn.Text = "Stop logging"           // Flagga för stopp
    TxtLogBtn.Text = "Start logging"              // Återställ flaggan
    LogTimer.Enabled = False                      // Stoppar loggningstimern
    ...
End If
```

Figur 3. Exempel på flaggor

4.4 Minitab

Minitab är en lättare version av programmet OMNITAB, utvecklat på Pennsylvania State University. Det är paket med statistikverktyg med vilket det utförs kvalitetskontroll, statistiska analyser, designa experiment och tolka data med olika grafiska hjälpmedel. Minitab användes för att designa ANOVA Gage R&R-testet. (Minitab Inc., 2010, s. 3–21)

4.5 ANOVA Gage R&R

En viktig del av arbetet var att utvärdera testprocessen efter att applikationen konstaterats fungera som önskat – slipmaskinen körs trådlöst i tre minuter och sluttemperaturen noteras för bedömning. För att bedöma om sluttemperaturen är acceptabel eller inte måste mätprocessen av temperaturen vara konsistent. Detta är orsaken till att det bestämdes att ett ANOVA Gage R&R-test skulle utföras.

ANOVA Gage R&R är ett analyseringssystem för mätprocesser som analyserar varians av mätdata när olika faktorer slumpmässigt påverkar mätningen. ANOVA är en förkortning av ”analysis of variance”, vilket betyder analys av varians. Varians ett mått på hur utspridd fördelningen av mätvärden är kring väntevärdet i en normalfördelning. Vid analysen belyses vilka faktorer som har mest påverkan och till vilken grad.

Osäkerhet i mätsystem är ett känt problem, ett sorts brus kan uppstå i data från mätningar. Det bör undersökas vilka faktorer som orsakar mest varians i data. Detta görs för att få veta om det är fel i tillverkningsprocessen eller om det kommer från variansen mellan användare. Mätsystem i sig själva har en viss osäkerhet; personen som utför mätningen kan vara en orsak till varians i resultatet, det som mäts kan ändra från gång till gång samt även omgivningen kan ändra.

R&R står för repeterbarhet och reproducerbarhet. Repeterbarhet är användarens förmåga att repetera processen med minsta möjliga varians i mätdata. Reproducerbarhet är mätsystemets förmåga att ge korrekt data från processen för mätningar gjorda av andra operatörer i andra omständigheter. Med ANOVA Gage R&R märks det om en viss operatörs resultat avviker från medelresultaten, eller om operatörernas resultat är likadana men testdelarna ger varierande resultat.

För att utföra detta användes Minitab, ett program med möjlighet att arrangera och tolka ANOVA Gage R&R-test. Det finns inget officiellt minimumkrav för antalet sampel men tio testdelar (till exempel slipmaskiner), två repetitioner och två operatörer anses räcka. För detta programs test användes endast fyra maskiner med olika komponenter, men tre operatörer och tre repetitioner per maskin, vilket borde ge en acceptabel bild av variansen. (ANOVA Gauge R&R, 2014)

5 Visual Basic .NET

Applikationen som användes som utgångspunkt för testapplikationen var skriven i programmeringsspråket Visual Basic .NET. Visual Basic .NET är ett högnivåspråk med flera programmeringsparadigmer, implementerat på .NET Framework. .NET Framework är en systemkomponent som är en del av operativsystemet Microsoft Windows. Det är ett slags bibliotek av komponenter som tar hand om exekveringen av program skrivna för detta ramverk. Till ramverket hör också ett klassbibliotek med förkodade lösningar för vanliga programmeringsuppgifter såsom kryptografi, databashantering, anslutning till nätverk, webbtjänster och algoritmer. Ett annat språk som också använder ramverket är Visual C#.

Visual Basic .NET är ett objektorienterat och händelsedrivet språk. Det använder kommandon för att beordra åtgärder, ofta är det uttryck och flaggor som ska utvärderas, för att sen kalla funktioner eller ändra värden på variabler.

Visual står för den grafiska delen av språket, vilket betyder skapande av fönster, knappar, grafer, timers och så vidare. För att skapa dessa krävs ingen programmering utan endast att de placeras i fönstret för applikationen som designas och behövs då bara ställas in. (Prosise, 2002, s. 27–35).

5.1 Objektorienterad programmering

Visual Basic är ett objektorienterat språk, till skillnad från tidigare språk som är processcentrerade. Processcentrerade språk börjar från toppen, går nedåt och definierar de variabler och funktioner som behövs vartefter. Om processen ändras bryts kedjan och det krävs många ändringar. Objektorienterad programmering vill undvika det här problemet med ett objekt-centrerat språk. Applikationen byggs upp av färdiga block av kod. Om processen ändras behövs bara vissa objekt ändras utan att kedjan bryts. Detta möjliggör snabbt och effektivt arbete. Om vissa kriterier följs kan vem som helst skapa ett block med kod, skicka det åt en annan person som då helt enkelt anropar det i sin applikation. (Prosise, 2002, s. 27–35).

5.2 Klasser

Alla objekt består av data och kod och fungerar som en enhet. Dessa objekt definieras som instanser av en klass. Klasser innehåller attribut, variabler, procedurer och händelser för ett objekt. Ett objekt skapas när en klass används och data sparas i den under ett namn. Om objektet skapas som en klass kan det användas i andra program som en del av .NET ramverket. Ett exempel på en klass är den stora basklassen System. Under System hittas System.Math, vilken är en underklass som innehåller matematiska beräkningar, till exempel den som beräknar kvadratrötter – System.Math.Sqrt. Inuti klasser sparas variabler precis som i andra programmeringsspråk.

```
Public class Application
{
    Private Sub Test
        public string    myString    = "Textsträng";
    End Sub

    Private Sub Test2
        public int        myInt      = System.Math.Sqrt(4);
    End Sub
}
End Class
```

Figur 4. Exempel på en klass.

Från klassen "Application" hittas Application.Test.myString vilket är textsträngen "Textsträng". Därifrån hittas också Application.Test2.myInt vilket är heltalet 2. Private Sub Test och Private Sub Test2 är två underklasser till Application. (Prosise, 2002, s. 27–35)

5.3 Timer

En timer är en klass som fungerar som en loop. Den kör enligt ett visst intervall som bestäms som ett attribut i objektet. I belastningsapplikationen användes en oändligt rullande timer. En timer är en klass med bland andra det Boolska attributet "Timer.Enabled" som kan vara True eller False. Med det kan den behändigt startas och stoppas ögonblickligen med ett kommando. (Prosise, 2002, s. 27–35)

6 Metoder och utförande

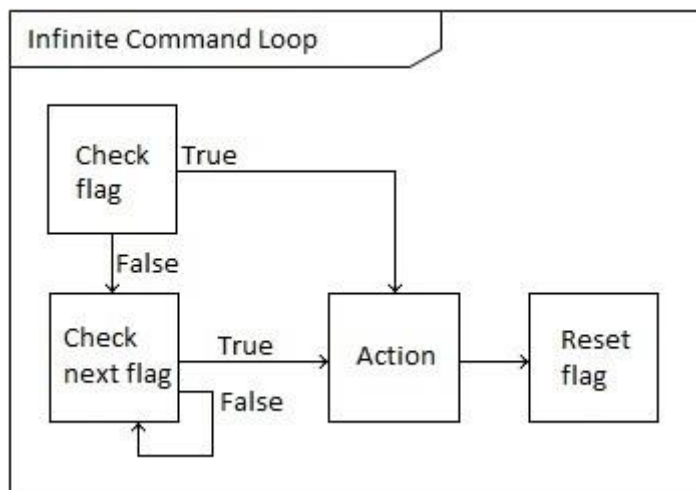
Arbetet inleddes med att lära sig behärska programmeringsspråket Visual Basic .NET. Tidigare applikationer öppnades i Visual Studio 2010 och deras kod analyserades. Detta var dock inte tillräckligt så guider på Internet söktes upp och med dessa inskaffades behövlig skolning för att kunna börja förstå språket. Även handledaren på Power Tools, Tomas Södergård, stod för en del rådgivning. Mycket tid ägnades åt att planera funktionsprincipen och testtiden. Testtiden önskades vara så kort som möjligt, men behövde också vara tillräcklig.

6.1 Påbörjande

Först skapades ett fönster för belastningsapplikationen, vilken benämndes "Heat Test" men ändrades senare till det mer korrekta "Load Test". I fönstret placerades ett par knappar vilka användes till att programmera de mest elementära uppgifterna, exempelvis att få fram textmeddelandet "Hello World". Till näst försöktes slippmaskinen fås igång från datorn via Bluetooth kommunikation. Det lyckades även om koden hade buggar, men kod- och programprinciper fungerade och byggdes på. Genom hela applikationen användes aldrig konstanta variabler för parametrar som kunde komma att behöva ändras på i ett senare skede. (Liew Voon Kiong, 2012)

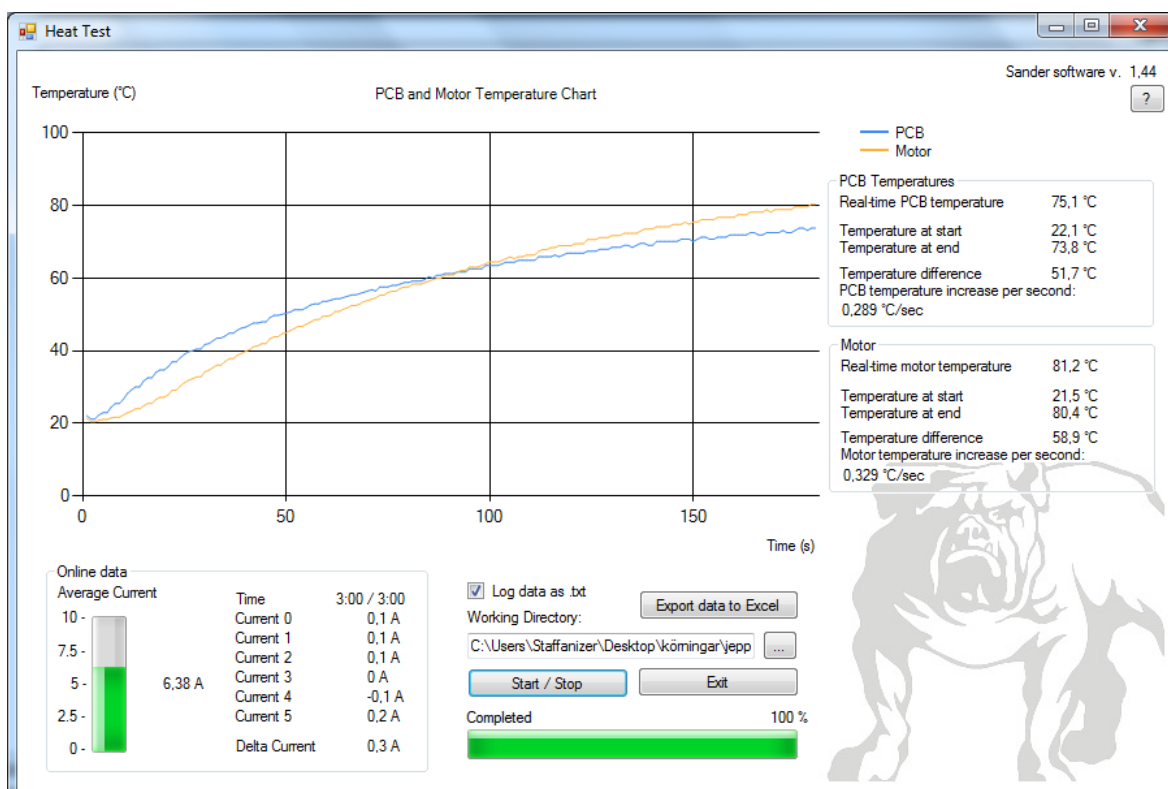
6.2 Funktionsprincip

Med tidigare erfarenhet av programmering bestående av endast korta kodsegment för matematikberäkningar var det en främmande situation att skapa ett helt program. Variabler och vektorer för datalagring deklarerades likt andra programspråk. För funktionsexekvering användes en timerloop ur ramapplikationen som kördes oavbrutet med 300 millisekunders intervall. Loopen var full av if-satser med flaggor som uttryck att evalueras. Vid påslagna flaggor stannade loopen för att köra funktionen som fanns i if-satsen, återställa flaggan och att sedan börja om från början. Med hjälp av flaggor hoppade funktioner mellan varandra i en bestämd följd och utförde kommandon, datalagringar, beräkningar med mera. (Liew Voon Kiong, 2012)



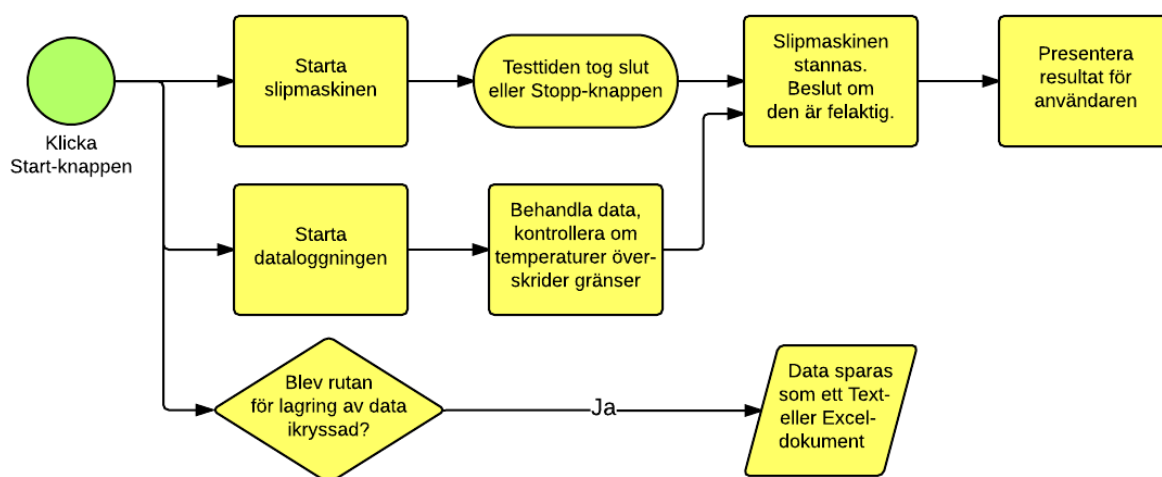
Figur 5. Blockschema för kommandoloopen

Figur 5 visar kommandoloopens utförande. Loopen består av en lista med if-satser som fungerar som flaggor för uttryck som ska evalueras. Vid påslagna flaggor stannar loopen för att köra funktionen som finns i if-satsen, återställa flaggan och sedan börja om från början. Vid nästa varv körs loopen till den första påslagna flaggan, utför funktionen och börjar om.



Figur 6. Heat Test, belastningstestapplikationen vid slutet av en körning. PCB temperatur betyder kretskortets temperatur, Printed Circuit Board. Namnet ändrades senare till det mer korrekta "Load Test".

När startknappen klickas startas slipmaskinen, samtidigt som en timerloop startas där realtidsdata sparas i vektorer en gång i sekunden. Alla strömmar och alla temperaturer sparas. När tre minuter gått stannas processen och en funktion jämför sluttemperaturen med gränstemperaturer och meddelar om den är inom tillåtna gränser. I fönstret visas temperaturökningen i realtid i en graf som ständigt uppdateras, även kretskortets och motorns temperaturer visas. (Liew Voon Kiong, 2012)



Figur 7. Flödesschema för testprocessen

6.3 Problem med Bluetooth-kommunikation

Ett större problem som påträffades hade att göra just med kommandoloopen. Flera kommandon skulle skickas till slipmaskinen och vid en flagga skrevs då in att fyra kommandon skulle skickas direkt efter varandra. Detta ledde till att inte alla kommandon kunde skickas. Det visade sig att slipmaskinen inte kunde ta emot fyra kommandon på en gång. Efter en tids felsökning misstänktes att slipmaskinen inte hann bearbeta kommandona på grund av Bluetooth-tekniken.

Bluetooth versionen som används på slipmaskinen är Bluetooth version 4, även kallad Bluetooth Smart. En funktion i Bluetooth-kommunikation går ut på att mottagaren sänder en bekräftelse på att den data som blivit överförd är fullständig och att inget har fallit bort under överföringen. Detta gör att totala överföringen blir långsammare men felfri. Denna funktion gavs inte tid att genomföras så vissa kommandon utelämnades.

Därmed skapades det en till timerloop i programmet bara för dessa fyra kommandon. När den startades skickade den kommandon med 100 millisekunders intervall varje gång den körde och då blev alla kommandon sända. Det skulle dock ha krävts att själv skapa Mirkas diagnostik-ramapplikation för att på förhand veta hur den presterar.

Nedan visas ett exempel på en timer där fyra kommandon sänds, ett per körning. Attributet 100 millisekunder användes, men syns inte i detta kodsegment.

```
Private Sub StartSanderTimer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles StartSanderTimer.Tick

    counter += 1

    Select Case counter
        Case 1
            FormBluetooth.RemoteActivate = True
        Case 2
            FormBluetooth.SanderOn = True
        Case 3
            FormBluetooth.HeatTestFullSpeed = True
        Case 4
            FormBluetooth.StartSander = True
            counter = 0
    End Select

End Sub
```

Figur 8. Exempel på en timer

Ett annat sätt att felfritt överföra data i stället för att vänta en viss tid, är att mottagaren meddelar när data blivit överfört, innan nästa mängd data sänds. På detta vis sker inga krockar. (Personlig kommunikation, Södergård, 1.8.2014)

6.4 Datatolkning och testtid

Innan kodningen kunde påbörjas behövde processen studeras för att få en aning om hur den beter sig. Temperaturen ökar snabbt i början och avtar långsamt för att stabiliseras efter tre till fyra minuter. Tiden för testet önskades vara så kort som möjligt för att effektivera processen med kontroll av stora mängder slipmaskiner. Dessutom finns det ingen ställning att sätta slipmaskinen i under testet, så personen som utför testet skulle vara tvungen att hålla i slipmaskinen medan den kör. Först övervägdes två minuters testtid men det fanns då ännu för stora variationer mellan slipmaskinerna. Efter tre minuter hade temperaturerna planat ut och gav en mer korrekt bild av förloppet.

Analysen av temperaturkurvan övervägdes länge. Olika kurvformer granskades för att söka efter karaktäristik som skulle underlätta tolkningen, men det visade sig att alla slipmaskiner hade exakt samma temperaturkurva med endast olika grader av stigning:

$$T(s) = \frac{K}{T \cdot s + 1}$$

Approximativt kan man säga att uppvärmningen är ett första ordningens system med en förstärkning K samt en tidskonstant T. På grund av maskinens relativt stora massa blir tidskonstanten tämligen stor eftersom det är mycket material som ska värmas upp.

Ett antal tillvägagångssätt för analys av temperaturdata övervägdes:

- Notera när derivatan av temperaturkurvan sjunker under en viss gräns då kurvan planar ut.
- Notera tiden det tar för slipmaskinen att nå en viss temperatur.
- Notera temperaturen efter en viss tid.

Derivator var ett självklart medel för att tolka kurvor och det undersöktes hur dessa kunde utnyttjas. Ett enkelt alternativ var att se när derivatan sjunker under en viss gräns, alltså när kurvan börjar plana ut igen.

Ett annat sätt vore att se hur lång tid det tar innan slipmaskinen har uppnått en viss temperatur, men det ansågs bakvänt. Alla slipmaskiner planar inte ut vid samma sluttemperaturer, så vissa kanske inte når upp till den temperatur som valts som gräns. Därför borde en temperaturgräns placeras tidigt, vid en temperatur som alla slipmaskiner når upp till. Dock vid en tidig tidpunkt kan inte fortsatt värmeutveckling förutspås med säkerhet. Därför kan två maskiner som når upp till temperaturgränsen samtidigt komma att plana ut vid olika sluttemperaturer, där den ena är överhettad.

Till slut valdes det enklaste sättet att se om en maskin överhettas eller inte – temperaturen noteras vid en tid då alla slipmaskiners temperaturer med säkerhet har börjat plana ut. Därefter jämförs temperaturen med medeltalet av andra slipmaskiners. Denna metod valdes eftersom den ansågs tillräckligt precis och att det vore onödigt att göra det mera komplicerat. Slipmaskinerna skulle vara tvungna att hamna inom bestämda temperaturgränser för att passera testet. Dessa gränsvärden kom att bestämmas senare, eftersom de bestäms genom testkörning av ett stort antal maskiner och med hjälp av normalfördelningen av temperaturerna. (Personlig kommunikation, Nordström & Södergård, 1.7.2014)

6.5 Felhantering, kompatibilitet och underhåll

Vid skapandet av applikationen måste möjliga felsituationer tas i beaktande. Om dessa lämnas obeaktade kan programmet låsa sig, eller operatören kan få fel data och då också fel bild av processen. Dessa förverkligades med flaggor som konstant kontrolleras. Det fanns egentligen endast en felsituation – om strömkabeln dras ut. Om detta händer ändras en flagga direkt, programmet avbryts och ett felmeddelande dyker upp.

När startknappen klickas kontrolleras däremot ett antal krav:

- Är slipmaskinen inkopplad?
- Är slipmaskinens mjukvaruversion tillräckligt ny?
- Är slipmaskinen varm från en tidigare körning?
- Har användaren kvitterat varningen om att slipmaskinen automatiskt börjar köra?

Alla dessa krav kontrolleras vid starten med hjälp av if-satser och frågesatser. Om inga flaggor är aktiverade startas testet. På grund av att testapplikationen bygger på en tidigare diagnostiseringsapplikation som har vissa begränsningar och är anpassad enligt vissa slipmaskiner och deras mjukvara, bör kravet på mjukvaruversion säkerställas innan programmet försöker interagera med slipmaskinen. Temperaturerna kontrolleras också i starten så att slipmaskinens vilotemperatur inte är för hög, för i så fall skulle den gå varmare än andra slipmaskiner och felaktigt anses avvikande. Belastningstestet går ut på att höja temperaturen, om den är hög från förr blir resultatet felaktigt. Sista kravet är en säkerhetsåtgärd, där användaren varnas för att slipmaskinen kommer att börja köras. (Personlig kommunikation, Södergård, 4.6.2014)

7 Användbarhet och användarvänlighet

Användbarhet och användargränssnitt var en viktig del av arbetet. Applikationen som designas är avsedd för människor världen över på Mirkas service center och inte nödvändigtvis bara ingenjörer. Arbetet med användbarhet och användarvänlighet är ett engångsarbete som har långvarig påverkan på applikationers användning.

7.1 Definitioner

Användbarhet definieras enligt ISO-normen 9241-11 som följande:

”Den grad i vilken användare i ett givet sammanhang kan bruka en produkt för att uppnå specifika mål på ett ändamålsenligt, effektivt och för användaren tillfredsställande sätt.”

Användarvänlighet är ett besläktat uttryck som handlar om hur lätt ett program är att använda, men säger ingenting om hur ändamålsenligt och effektivt det är. Testapplikationen skulle vara både användbar och användarvänlig. Med det avses att applikationen ska vara rakt på sak, allting som behövs ska visas i applikationsfönstret och allt annat av sekundärt intresse ska döljas. Programmet ska vara lätt att lära sig och lätt att minnas. Programspråket blev engelska eftersom det ska kunna förstås världen över.

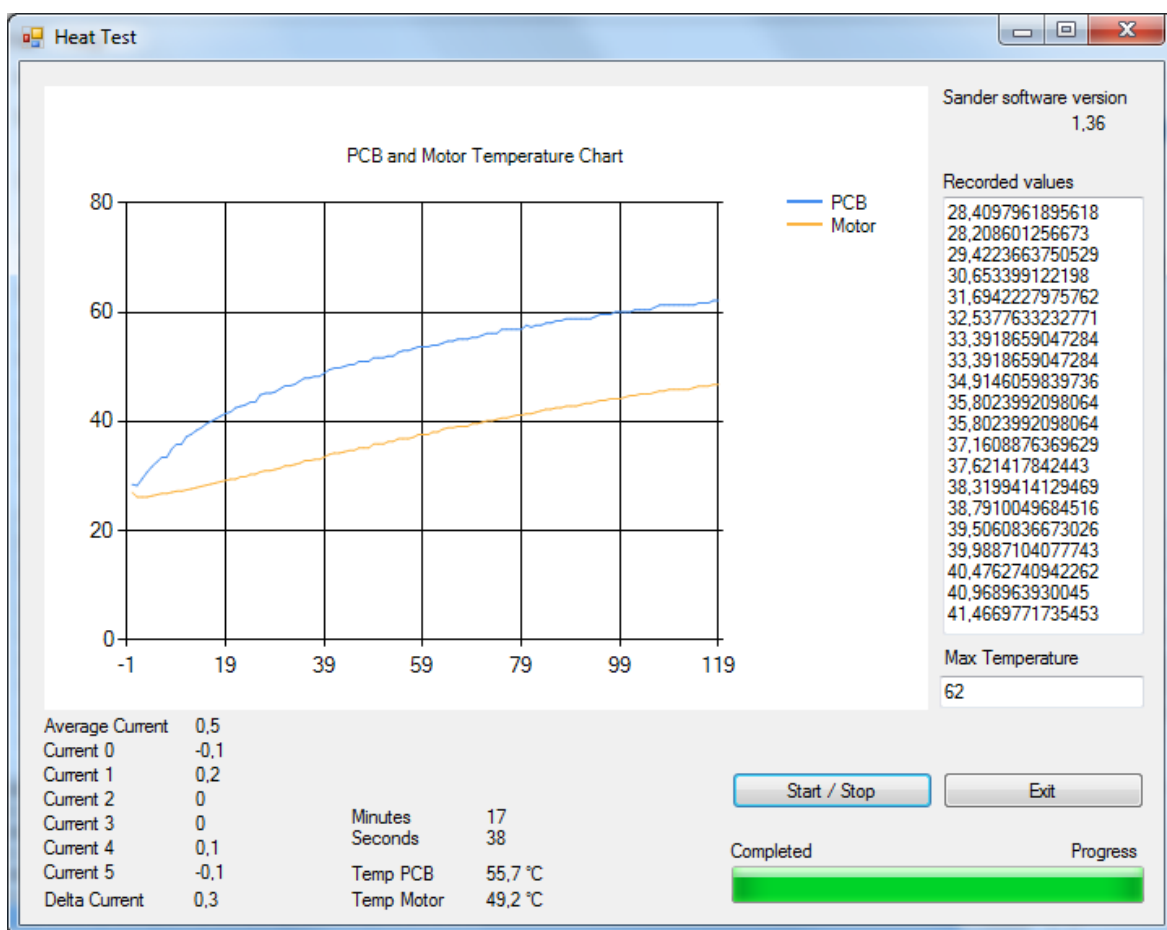
Enligt ISO 9241-12 finns sju attribut som bör beaktas vid design av grafiska gränssnitt:

- Klarhet: informationen är överskådlig och precis.
- Urskiljningsförmåga: den uppställda informationen kan bli urskild korrekt.
- Koncishet: användare är inte överösta med ovidkommande information.
- Konsistens: en unik design, överensstämmelse med användares förväntningar.
- Spårbarhet: användarens uppmärksamhet är riktad mot information i fråga.
- Läslighet: informationen är lätt att läsa.
- Begriplighet: betydelsen är tydlig att förstå, otvetydig, lätt att tolka och igenkännbar.

Dessa attribut togs i beaktande vid designen av användargränssnittet. Några punkter kommer att behandlas mer ingående. (International Organization for Standardization, 1998, ISO 9241-11).

7.2 Klarhet och spårbarhet

Programmet blev designat så att det enda användaren behöver göra är att trycka på startknappen och kvittera att denna förstått att maskinen börjar köra, så sköter programmet resten. Programmet blev väldigt enkelt och användarvänligt. Startknappen fungerar också som stoppknapp om maskinen redan kör. Detta minskar på antalet knappar att hålla reda på och är också ett naturligt val. Valet att grafiskt visa temperaturökningen gjordes för att användaren ska kunna se händelseförloppet, vilket är väldigt användbart för att kunna observera trender i data. Användaren kan följa med temperaturerna både i exakta nummer och på grafen. Vid extremtillfällen, såsom om en slipmaskin kommer att överhettas problematiskt mycket, kan dessa visuella hjälpmedel varna användaren så att testkörningen kan stoppas i tid.



Figur 9. Figuren visar en bristfällig tidig version av det grafiska gränssnittet.

Strömmarna på motorns alla faser uppdateras också i realtid med en sekunds intervall. Med dessa har användaren möjlighet att observera om mängden ström är acceptabel. Förutom

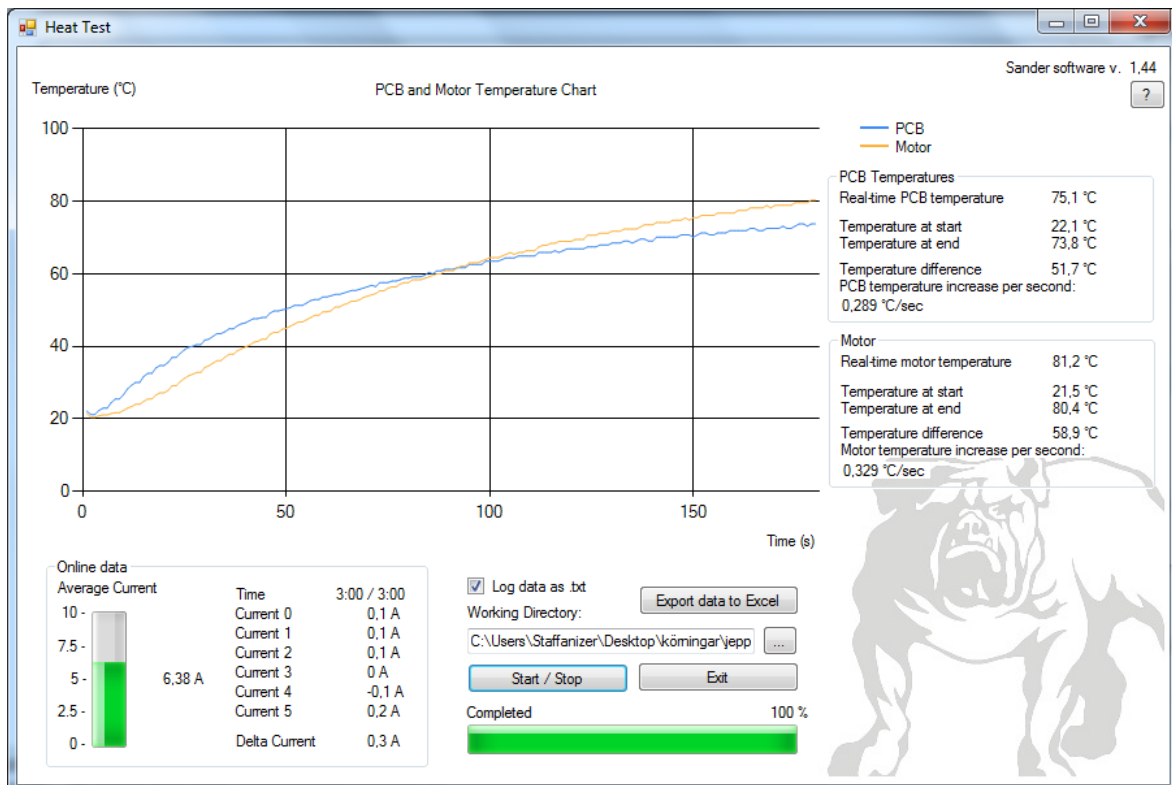
dessas visas även deltaströmmen, vilken är strömmen mellan olika faser och borde vara så låg som möjligt. Därtill visas tiden testet har varit igång och hur länge det är kvar, vilket också framställs grafiskt med hjälp av en växande grön statusbar.

En tidig version av applikationen innehöll all relevant data men var inte användarvänlig (Figur 9.). Data var inte uppordnat för att lätt kunna åskådas, en lista med sparade temperaturvärden som skulle ha gjort användaren förundrad visades helt i onödan, grafen var oläslig och så vidare. Detta var en halvfärdig version som visades åt handledaren. Det som anmärktes var att konsistens borde tas i beaktande. (International Organization for Standardization, 1998, ISO 9241-11).

7.3 Konsistens

För att skapa mera konsistens i designen studerades ramapplikationens utseende. Det första som märktes var en vit bakgrund med en bulldogg i hörnet, Mirkas logo. Den bakgrunden var det första som kopierades till testprogrammet. Därefter märktes det att informationen inramas i boxar med rubriker, vilka håller ihop data och gör utseendet mer estetiskt. När det också kopierats började programmet se mycket trovärdigare ut. Listan med sparade temperaturvärden slopades helt eftersom det var överflödigt data, det enda som räknas är sluttemperaturen. Trender i temperaturlistan kan ej heller enkelt läsas ur siffror utan det står grafen för. För "average current", alltså genomsnittsström, skapades en stående statusbar. Den fanns inte i Visual Basics standardkomponenter så den hämtades från internet. Valet att använda en statusbar för genomsnittsströmmen är för att användaren direkt ska se trender och onormala ökningar vilka kan vara till fara för slipmaskinen. Att endast visa dessa i nummer kan göra att de lätt går osedda.

Utseende är en viktig del av programdesign för det är det visuella som först bedöms, om programmet ser bra ut är användaren genast nöjd. Användbarhet är också visuellt till stor del, det är viktigt att programmet fungerar men lika viktigt att det är lätt att navigera och tolka.



Figur 10. Testapplikationen efter att konsistens som designprincip tagits i beaktande.

Programmet skulle göras så klart och koncist som möjligt. Till en början fanns två knappar vilka räknade ut genomsnittstemperatur och mediantemperatur över hela testtiden, men vid närmare eftertanke ansågs det överflödigt så de togs bort. En hjälpknapp placerades uppe i högra hörnet ur vägen med instruktioner för hur programmet används.

Strömmar placerades i en egen box och så även kretskorts- och motortemperaturerna. Nu började allting vara klart och alla attribut uppnådda. (International Organization for Standardization, 1998, ISO 9241-12).

8 ANOVA Gage R&R testresultat

Med testprocessen gjord var nästa steg att göra en mätsystemanalys med ANOVA Gage R&R. ANOVA Gage R&R-testet gick till så att en person i gången kör alla fyra maskiner i belastningstestet och sluttemperaturerna noteras. Ordningen maskinerna körs i slumpas fram på nytt varje gång operatören byts. Detta görs för att minimera inverkan av mänskliga faktorer som till exempel trötthet och koncentration. Efter varje persons körning av maskinerna behövde maskinerna tre timmar på sig att kallna innan de kunde köras igen. Därför tog denna utvärdering flera dagar att utföra. Det gjordes nio körningar av tre olika operatörer med maskiner med olika typer av komponenter. (Minitab Inc., 2010, s. 3–21)

8.1 Resultat

Efter testet kunde det konstateras att maskinerna hamnar inom ett visst intervall varje gång, vilket betyder att belastningstestet endast har en liten varians och att programmet således är exakt och fungerar som önskat.

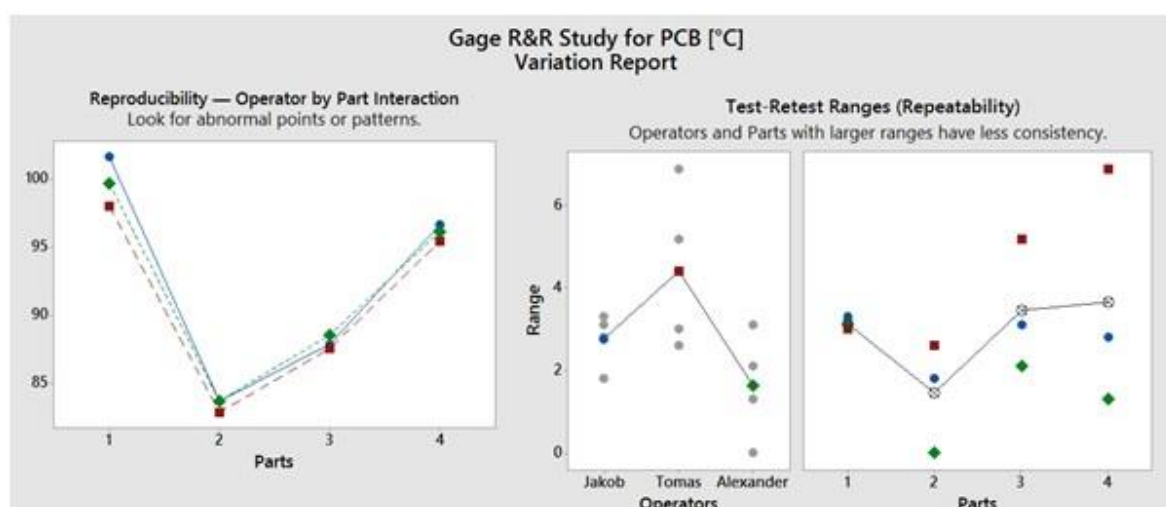
Operatör	Maskin 1	Maskin 2	Maskin 3	Maskin 4
Jakob	100,0	82,8	86,5	95,7
Tomas	100,0	83,7	90,7	100,0
Alexander	98,5	83,7	87,5	95,7
Jakob	103,3	84,6	89,6	98,5
Tomas	97,0	81,1	86,5	93,1
Alexander	98,8	83,7	88,5	95,7
Jakob	101,6	83,7	87,5	95,7
Tomas	97,0	83,7	85,5	93,1
Alexander	101,6	83,7	89,6	97,0
Medeltal	99,76	83,41	87,99	96,06

Figur 11. En tabell med resultatet av ANOVA Gage R&R testets mätvärden, här visas endast krets-kortens temperaturskillnader. Temperaturerna mättes i grader Celsius.

Operatör	Maskin 1	Maskin 2	Maskin 3	Maskin 4
Jakob	91,0	83,6	89,0	87,1
Tomas	92,1	85,3	94,3	90,0
Alexander	91,0	84,4	91,0	87,1
Jakob	94,3	86,2	92,1	89,0
Tomas	88,0	81,9	90,0	83,6
Alexander	90,0	84,4	91,0	86,2
Jakob	91,0	83,6	90,0	85,3
Tomas	87,1	84,4	88,0	84,4
Alexander	92,1	84,4	93,2	86,2
Medeltal	90,73	84,24	90,96	86,54

Figur 12. En tabell av temperaturerna hos motorerna från maskinerna i tabellen i Figur 8.

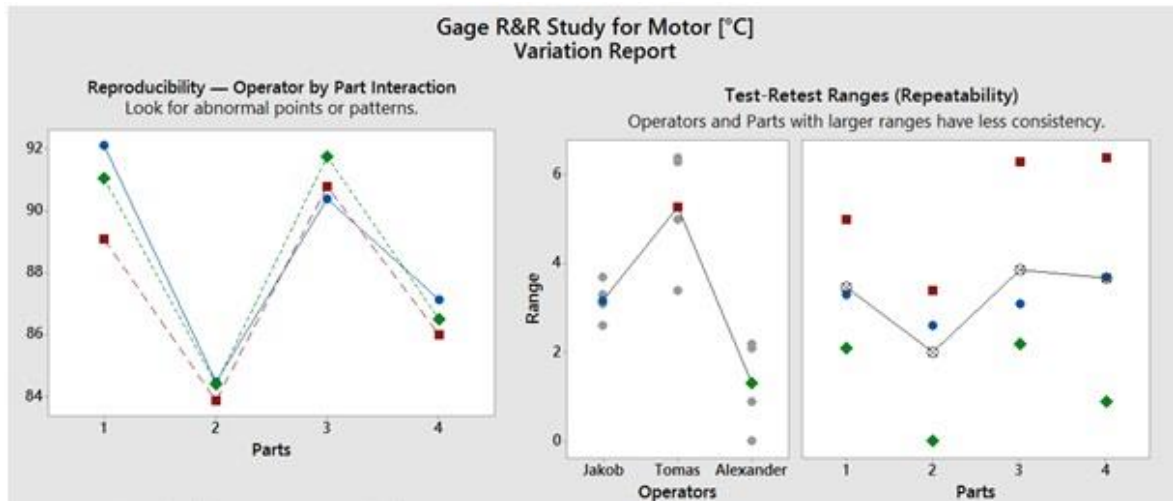
Maskin 1 och 4 var oinkörda maskiner. I tabellen i Figur 8 märks att deras kretskort når mycket högre temperaturer än de andra, de blir tio till femton grader varmare. Detta torde bero på att slipmaskinernas lager inte var innötta och går således trögare och genererar mera värme.



Figur 13. Här visas en bild av analysen med Minitab-programmet.

Graferna i Figur 13 visar variationen i kretskortstemperaturer med tre olika operatörer, samt kretskortens varierande temperaturer under flera körningar med samma operatör. Från Figur 13 under "Reproducibility", reproducerbarhet, visas medelsluttemperaturer för slipmaskinerna. Ingen stor variation syns mellan operatörer, endast mellan olika slipmaskiner. Detta är bra, stor variation är oönskad och betyder att testet är opålitligt. Slipmaskinerna har dock olika komponenter så variation sinsemellan dem är naturligt.

I samma figur under ”Test-Retest Ranges (Repeatability)”, repeterbarhet, visas att det finns en viss variation både inom samma operatörs körningar samt mellan olika operatörer. Att det varierar mellan operatörer kan bero på små faktorer i körordningen, som att Södergård, handledaren, ofta körde maskinerna som andra operatör för dagen och då kanske maskinerna inte hade hunnit kallna tillräckligt.



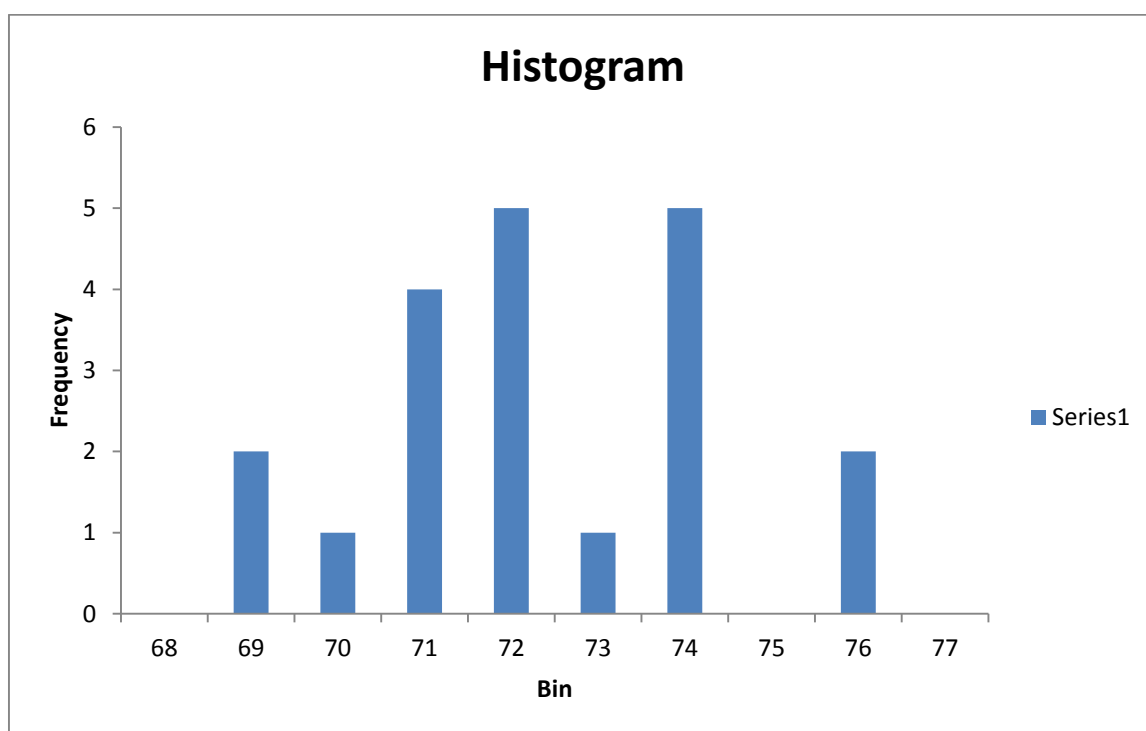
Figur 14. Här visas variationen i motortemperaturer med olika operatörer, samt slipmaskinmotorernas varierande temperaturer under flera körningar med samma operatör.

I Figur 14 visas reproducerbarhet och repeterbarhet för temperaturerna hos respektive slipmaskiners motorer. Här syns lite större variation mellan operatörer, men motorer har också större massa som värms upp under körningen så andra resultat kan förväntas.

Från ANOVA Gage R&R-testet konstaterades att slipmaskinerna inte har alltför stor variation mellan sina egna körningar och testet därför är tillräckligt precist och kan användas. Variationerna som finns förklaras med att maskinerna har väldigt varierande komponenter, de hade olika höljen och olika motorer. (Minitab Inc., 2010, s. 3–21)

9 Gränsvärden

För att bestämma temperaturgränsvärden för testprogrammet behövdes data införskaffas om de egentliga slipmaskinerna. Efter ANOVA Gage R&R testet kördes därför 20 nya oanvända slipmaskiner i belastningstestet. Temperaturfördelningen antogs genast vara normalfördelad eftersom slipmaskinerna massproduceras och görs så likadana som möjligt. Genom att avbilda temperaturerna med hjälp av ett histogram kunde detta bekräftas. Det märktes att temperaturerna hölls just över 70 grader i huvudsak, med ett medelvärde på 71,9 °C. Detta var i stor kontrast till maskinerna som användes i ANOVA Gage R&R testet. De nya maskinerna har nya höljen med effektivare design för luftkylningen.



Figur 15. Histogram på temperaturerna för kretskorten på 20 oinkörda slipmaskiner samlade runt 72 – 73 grader.

För att bestämma temperaturgränsvärden provades först t-fördelning. En t-fördelning är en sannolikhetsfördelning som används för att testa signifikans för undersökningar med små populationer. I detta fall användes t-fördelningen för att räkna ut temperaturgränser, konfidensintervall, till belastningstestet. Konfidensintervall säger med en vald konfidensgrad inom vilket intervall ett godtyckligt nytt mätvärde hamnar. För att kunna använda t-fördelning behövs två värden, antal frihetsgrader och konfidensgrad. T-fördelning bestämmer att med 20 testobjekt fås 19 frihetsgrader. Konfidensgraden valdes till 97,5 %. Enligt de värdena fås t-faktorn 2,093 ur en tabell. T-faktorn är en statistika, ett tal som beskriver urvalet och som används för att göra en skattning av en okänd parameter, i detta fall konfidensintervallet. För att bestämma det behövdes först standardavvikelsen uträknas.

Standardavvikelsen är ett statistiskt mått på hur mycket de olika mätvärdena avviker från medelvärdet och fås med ekvationen:

$$S_n = \sqrt{\frac{\sum (x - m)^2}{n - 1}} \quad S_n = \sqrt{\frac{\sum (x - 71.9)^2}{20 - 1}}$$

$$S_n = 2.008$$

Figur 16. Ekvation för att bestämma standardavvikelsen och dess uträkning för kretskortet (Mattecentrum, 2014).

- x är alla observationsvärden och m är medelvärdet 71,9 °C
- $x - m$ är avvikelsen från medelvärdet
- I täljaren finns summan av alla kvadrerade avvikelser
- I nämnaren finns antalet sampel minus ett

Övre och nedre gränsvärdena fås från denna ekvation:

$$\bar{X}_n \pm A \frac{S_n}{\sqrt{n}} \quad a = 71.9 - 2.093 \frac{S_n}{\sqrt{20}} \quad b = 71.9 + 2.093 \frac{S_n}{\sqrt{20}}$$

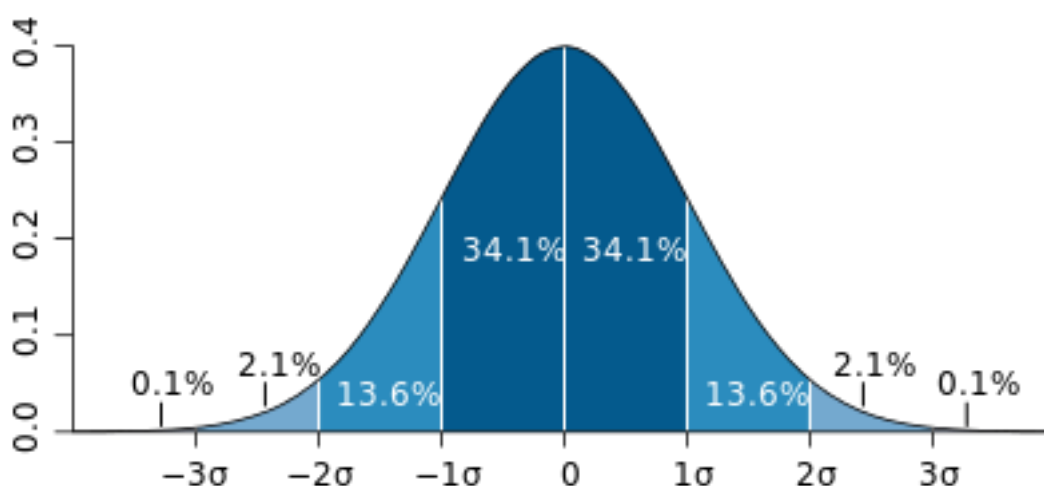
$$a = 70.96 \quad b = 72.84$$

Figur 17. Ekvation för bestämning gränsvärden och dess uträkning (Mattecentrum, 2014).

- \bar{X}_n är medeltalet för alla körningar
- A är t-faktorn
- S_n är standardavvikelsen
- n är sampel mängden

Genom att sätta in värden och lösa ekvationerna fås 70,96 – 72,84 °C som gränsvärden. Detta svarar endast på frågan inom vilket intervall populationens medelvärde bedöms vara, men ansågs ändå för snävt. Genom en snabb koll på histogrammet märks det att flera värden faller utanför.

Därför provades ett annat sätt, Chebyshevs olikhet, enligt handledarens förslag. Det säger att 75 % av alla sampel faller inom två standardavvikelser från medeltalet, eller att 88,89 % faller inom tre standardavvikelser. Enkel addition och subtraktion visade att alla uppmätta värden på sluttemperaturen var inom två standardavvikelser från medeltalet.



Figur 18. Normalfördelning. 0 står för väntevärdet μ , σ står för en standardavvikelse. Väntevärdet och medelvärdet är i detta fall likadana.

Två standardavvikelser under medelvärdet ger 67,88 °C för kretskortet medan två standardavvikelser över ger 75,92 °C. De uppmätta värdena 69 till 75,80 °C faller inom detta intervall. Marginalerna konstaterades vara goda och skrevs in i koden för belastningstestet. Intervallet utökades med två grader åt båda hållen för att inte vara onödigt strängt.

Detta intervall gäller dock endast för slipmaskinsmodellen MIRKA® DEROS 5650CV som inte blir lika varm som äldre modeller med annat hölje. Om man kör testet med detta intervall på äldre maskiner passerar de knappast testet. Dessutom borde man kunna anta att majoriteten av alla tidigare producerade slipmaskiner är godtagbara, så testet bör kalibreras om vid körning av äldre maskiner. (Mattecentrum, 2014; Personlig kommunikation, Södergård, 31.3.2014)

10 Resultat

Analysen i applikationen kunde ha gjorts med många olika metoder men den enklaste metoden valdes till slut. Fördelen med enkelhet är att det fungerar utan problem. Behövs gränsvärdena ändras är det lätt att snabbt gå in i koden utan att vara desto mer insatt.

Att det satsades tid på den grafiska utformningen ledde till ett användarvänligt och användbart verktyg. Statusbaren för medelström och i synnerhet den levande grafen kan varna användaren om slipmaskinen håller på att överhettas, vilket gör att testet kan stoppas och felaktiga komponenter bytas ut utan att både kretskortet och motorn brinner upp. Vid en uppgradering av testapplikationen kunde medelström visas i samma graf på en andra y-axis för att göra det ännu tydligare.

Utformningen på textloggningens utdata kunde ha gjorts mer beskrivande med mer information, såsom slipmaskinens serienummer och modell. Tiden för projektet tog slut före detta infördes, dock blev alla specifikationer för applikationen uppfyllda. Belastningsapplikationen fungerar som tänkt och utan problem.

Tiden för projektet kunde ha varit kortare. Jag spred ut arbetsdagar över tre månader, medan totala arbetsmängden var cirka två månader.

Utöver belastningsapplikationen gjordes en detaljerad användarmanual med information om hur slipmaskinen används, hur fläkten monteras fast, hur programmet används och många säkerhetsföreskrifter.

11 Diskussion

Jag hade aldrig använt Microsoft Visual Basic .NET förut, men det var likt C++ på många sätt, vilket jag var van med. Detta underlättade inläringen men det tog ändå en månad innan det började kännas under kontroll.

Det tog en lång tid i början av projektet innan jag fick på klart hur belastningsapplikationen skulle fungera. Eftersom jag inte har gjort fullständiga program i Visual Basic .NET hade jag ingen aning om hur de utformas, till exempel att oändliga loopar används. Även efter programmeringen började löpa blev jag tvungen att ägna en vecka åt att söka buggar i applikationen. Här var det igen brist på erfarenhet som skapade extra arbete.

ANOVA Gage R&R tog många dagar att utföra när maskinerna var tvungna att kallna. Resultatet av testet blev otydligt för maskinerna som användes var alltför olika. Detta på grund av för lite förberedelse. De fyra maskinerna hade fyra olika kretskort och motorer, två olika interna fläktar, två olika höljen och dessutom var två oinkörda medan två var inkörda. Det var svårt att veta hur komponenternas samverkan påverkade arbets-temperaturen.

Besöket vid Mirka i Jakobstad för att köra slipmaskiner i testprogrammet gav en värdefull samling av data som blev till stor hjälp vid bestämning av gränsvärdena. Denna data kan Mirka troligtvis använda även på andra sätt. Antalet slipmaskiner ansågs tillräckligt, men kunde också ha varit dubbelt eller tre gånger större för att få mer noggrannhet.

Framtida utmaningar för belastningsapplikationen är många. Testresultaten kan variera beroende på omgivningstemperaturen, årstider eller land och klimat. Därför kan temperaturgränserna behöva anpassas enligt testlokalen. Temperaturökningen i slipmaskinerna är inte linjär, så det kan också spela en roll vid större temperaturskillnader. Däremot är en fullständigt linjär temperaturökning troligen inte realistisk.

Det gav mycket att få arbeta på ett företag med en handledare. Jag har lärt mig mycket om Visual Basic .NET och ANOVA Gage R&R och tar gärna på mig sådana arbeten i framtiden.

Källhänvisning

Bluetooth SIG, Inc., 2015, *Bluetooth Basics* [Online]

<http://www.bluetooth.com/Pages/Basics.aspx> [Hämtat 15.1.2015]

Minitab Inc., 2010. *Gage Studies for Continuous Data*. [Online]

<http://www.minitab.com/uploadedFiles/Documents/sample-materials/TrainingSampleMeasurementSystemsMTB16EN.pdf> [Hämtat 15.1.2015]

International Organization for Standardization. *ISO 9241-11*: 1998. [Online]

http://en.wikipedia.org/wiki/ISO_9241 [Hämtat 15.1.2015]

International Organization for Standardization. *ISO 9241-12*: 1998. [Online]

http://en.wikipedia.org/wiki/ISO_9241 [Hämtat 15.1.2015]

KWH Group Ltd., 2009. *The History of the KWH Group*. [Online]

<http://www.kwhgroup.com/English/History> [Hämtat 15.1.2015]

KWH Mirka Ltd., 2013. *Elektriska excenterslipmaskiner*. [Online]

http://www.mirka.com/sv/fi/MASKINER/Fi-SE_Electric-Random-Orbital-Sanders/ [Hämtat 15.1.2015]

Mattecentrum, 2014. *Standardavvikelse*. [Online]

<http://www.matteboken.se/lektioner/matte-2/statistik/standardavvikelse> [Hämtat 4.2.2015]

Microsoft Corporation, 2015. *Objects and Classes*. [Online]

[http://msdn.microsoft.com/en-us/library/527aztek\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/527aztek(v=vs.90).aspx) [Hämtat 15.1.2015]

Prosise, J., 2002. *Programming Microsoft .NET*. Redmond: Microsoft Press s. 27 – 35

Valtanen, K. 2013. *Tekniikan taulukkokirja*. Mikkeli: Genesis-Kirjat Oy. s. 135 – 155

Liew Voon Kiong, 2012. *Visual Basic Made Easy*. [Online]

<http://www.vbtutor.net/index.php/visual-basic-2010-tutorial/> [Hämtat 15.1.2015]

ANOVA Gauge R&R, 2014. [Online]

http://en.wikipedia.org/wiki/ANOVA_gauge_R%26R [Hämtat 19.1.2015]

Student's t-distribution, 2015. [Online]

http://en.wikipedia.org/wiki/Student%27s_t-distribution [Hämtat 21.2.2015]

Chebyshev's inequality, 2015. [Online]

http://en.wikipedia.org/wiki/Chebyshev%27s_inequality [Hämtat 28.1.2015]

Södergård T., Nordström C., Personlig kommunikation, 2014.

Anställda på avdelningen Power Tools vid KWH Mirka Ab.



After-sales “Load Test” Application

Created 02.09.2014 by JST





Contents

1. Introduction.....	3
2. Setup.....	3
2.1 Mount the fan.....	3
2.2 Connect the devices.....	3
3. Application description.....	4
4. Testing the sander.....	5
5. Conclusion.....	5

1. Introduction

The main reason for this application is to standardize a temperature testing system and environment for Mirka Electric Sanders for after-sales control. MIRKA® DEROS 5650CV is the only sander tested with this application. The application runs the sander for a time and says if it's overheating or not.

2. Setup

2.1 Mount the fan

Remove the Backing Pad using a 24 mm wrench. Place the cooling fan and tighten it securely. If the cooling fan is broken in any way, for example a crack, then don't use it and find a new one instead. Look also if it's relatively clean. Any dirt in it may be launched from it when it starts spinning.

2.2 Connect the devices

First, insert a Bluetooth Dongle and start Mirka Power Tools Diagnostics. Connect to it by clicking the button "Select BT-dongle". Second, plug the power cord to MIRKA® DEROS 5650CV. Now two green led lights should be lit on the sander. Click the button "Connect to selected". Now you're connected to the Sander and ready to test it. Third, go to "File" and select "Load Test".

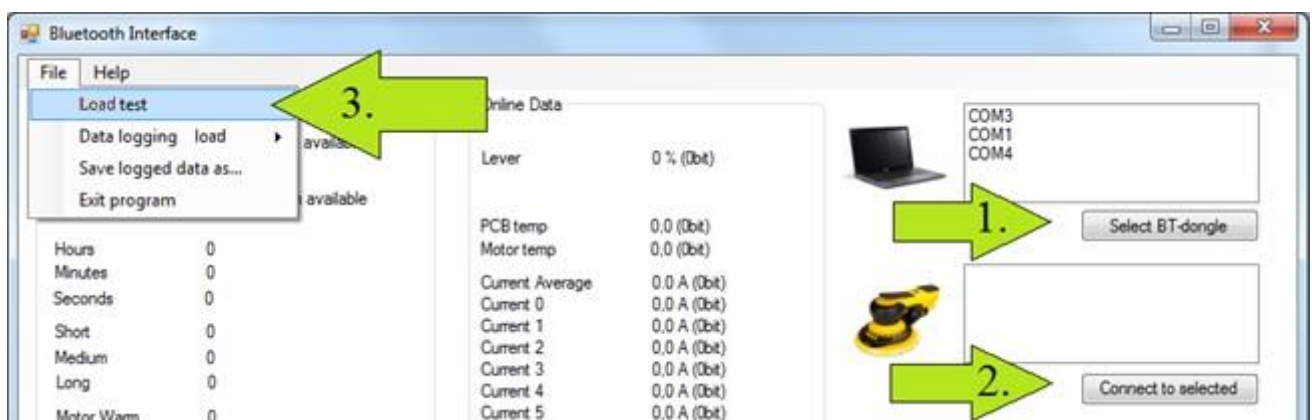


Figure 1. Shows the setup process

3. Application description

The application consists of a temperature/time-chart, a couple operating buttons and info on currents and temperatures. To the right one can see PCB and Motor temperatures. Don't confuse the "real-time temperatures" with "temperatures at the end". "Real-time PCB temperature" always shows the temperature at the moment even when the test is not running, "Temperature at the end" shows the last and usually highest temperature from the test. In the lower left corner all six currents over the six coils of the motor can be seen. Under those the Delta current is shown.

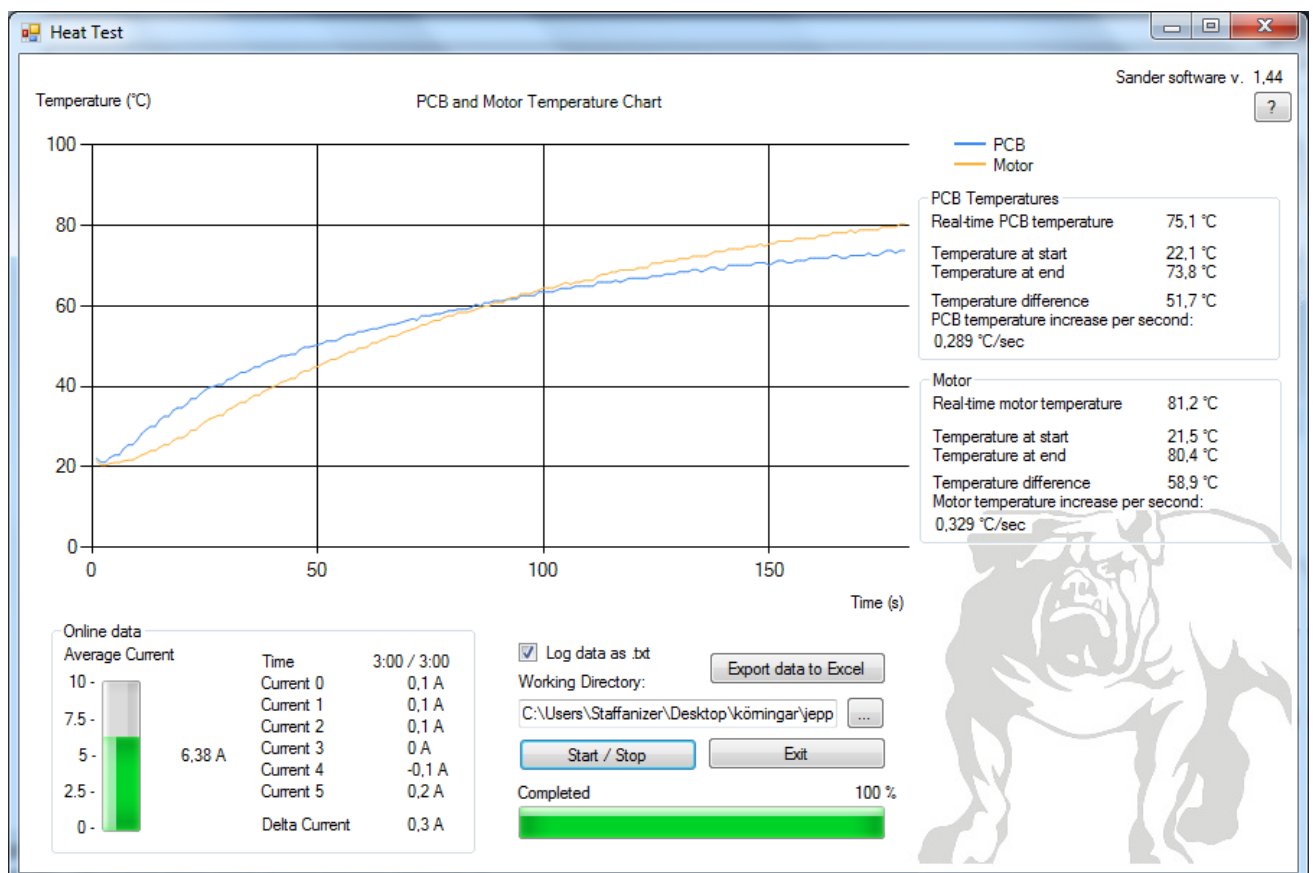


Figure 2. The PCB and motor temperatures are collected. The picture shows a completed run.

4. Testing a sander

First make sure it's set up according to step 3, above. Check the box "Log data as .txt" if you want a Text document displaying the results. After the testing is completed to 100 % the button "Export data to Excel" will be enabled. If you want to export it to Excel, which is perhaps more readable, click the button and choose a filename. NOTE – you can't save the data to Excel unless you had logged the data from the start, as in checking "Log data as .txt".

Hold the sander in the air. Keep the plane of the spinning disc away from your eyes in case pieces of dirt are launched from it. Wait three minutes until the test is completed. When completed, the sander will stop running and a message box will appear. In the message box there will be results from the test saying if the sander passed the test or not.

The delta current can touch 0,5A but it should stay under it. If it often is over 0,5A then the test will fail the sander.

5. Conclusion

If the test says the sander passed, the sander is OK.